(54) Title: ELECTRONIC DOCUMENT SYSTEM
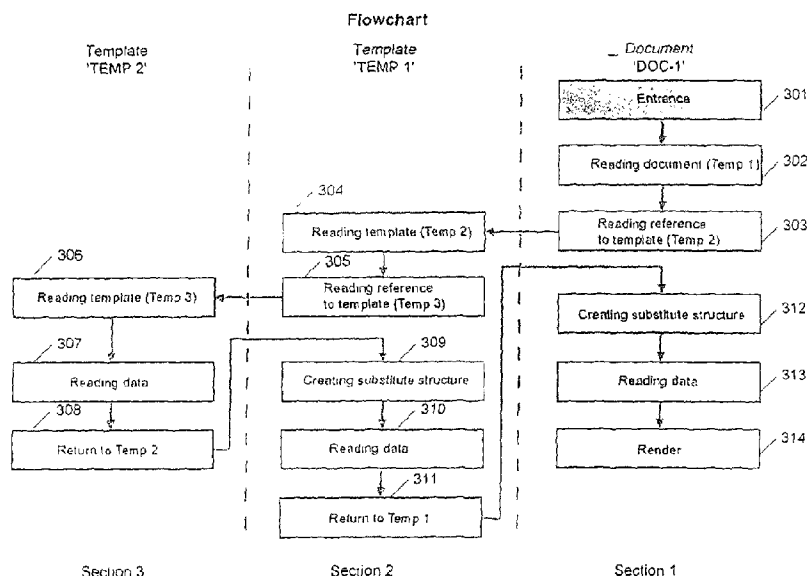
(57) Abstract: This invention related to a method of generating a multimedia document from a plurality of templates, wherein each template has an identifiable object with properties. The method comprises the steps of: searching a reference to a first template in an electronic document; parsing the first template to find a reference to a second template; parsing the second template to find a first object with properties; reading the properties, and returning to the first template; continuing parsing the first template to find a second object that specifies changes to the first object and change properties of the first object according to properties in the second object specifying the changes; and rendering the first object. Preferably XML, XSL, and/or XSLT languages are used for implementation.

Electronic document system

This invention relates to processing of an electronic document.

5   Electronic devices are increasingly being arranged to connect to the Internet. Devices that can access the Internet comprise Personal Computers, Laptop computers, Mobile telephones, Personal Digital Assistants (PDAs), and also household appliances such as Refrigerators, set-
10   top boxes for TV-sets etc.

The electronic devices are arranged to host some type of application program that provides for accessing information via the Internet. Since the application can take various forms depending on the device hosting the
15   application and even may be distributed across several devices and contexts there is a need for more flexible applications tailored to a hosting device or so-called application platform.

Generally, when creating applications it is often the
20   case that an application should be almost the same as a second application with just small but significant changes, that is what can be called "Same but different".

The languages Extensible Mark-up Language (XML) in combination with Extensible Style sheet Language (XSL)
25   and/or Extensible Style sheet Language Transformations (XSLT) are languages for representing data and presentation style/transformation rules for the data separately. The languages are specified by the www-consortium (http://www.w3.org/Style/XSL/). Application
30   programs utilizing these languages can thus use an XSL

style and/or transformation for making a presentation of data in XML format tailored to a specific platform.

However, known methods implementing XSL/XSLT are not flexible because they, in the presentation/transformation
5   only allow reuse of the data. Note, that the data are arranged in an XML structure.

This and other problems are solved when performing a method of generating a document from a plurality of templates, wherein each template has an identifiable
10. object with properties; the method comprising the steps of: finding a reference to a first template in an electronic document; parsing the first template to find a reference to a second template; parsing the second template to find a first object with properties; reading
15   the properties, and returning to the first template; continuing parsing the first template to find a second object that specifies changes to the first object and change properties of the first object according to properties in the second object specifying the changes;
20   and rendering the first object.

Thereby a system/technology that enables XML to be transformed/changed/generated without the need to know the resulting format and/or content, and with the possibility to reuse both data and structure of the
25   generated document, by expressing only the changes in respect to both data and structure.

Thus the above phrase "Same but different" is accomplished by expressing changes only. If a "same but different" approach of the data should be implemented in
30   XSL the full rule and structure must be re-specified, therefore XSL can only reuse the Data and not the structure of the generated document.

Preferably, the first object is rendered in accordance with the changed properties.

The properties to be changed of the first object are specified by similar properties in the second object. Thereby, changes can be identified and carried out expediently.

When the method further comprises the step of: constraining properties of objects such that the properties cannot be modified, high-level rules can be specified without modifying the individual parts of the templates. Thus some general policy can be enforced in a simple way.

When the method further comprises the step of: inserting further objects into the document in response to a user interaction, changes in structure and data can be generated by an editor recording changes through user interactions.

The method can comprise the step of copying a document and referenced templates into a set of files. Even when a document is created as a collection of other smaller parts from many different locations/contexts, and each of those parts have some dependencies of other parts, a resulting document chain can be used as a single object where all external dependencies are known and resolved into local references.

The method can comprise the step of merging a document and referenced templates into a single file. Even when a document is created as a collection of other smaller parts from many different locations/contexts, and each of those parts have some dependencies of other parts, the resulting document can be used as a single object where

all external dependencies are known and resolved into the parent document.

In a preferred embodiment the objects comprise multimedia objects.

5   The document and the templates can be written in a mark-up language. The generated mark-up document is "merged and changed" rather that "processed and generated" this enables reuse of both document structure and document data.

10   The invention will be explained below in connection with a preferred embodiment and with reference to the drawing, in which:

fig. 1 shows the composition of three different documents based a common template and three different templates,
15   respectively;

fig. 2 shows a combination of more templates;

fig. 3 shows a flow chart for parsing a document based on Three templates;

fig. 4 shows a pseudo flowchart illustrating the process
20   of parsing a document based on two templates; and

fig. 5 shows a computer system according to the invention.

Fig. 1 shows the composition of three different documents based a common template and three different templates,
25   respectively. A common template 101 comprises an object 102 e.g. in the form of a picture element. The object 101 comprises an identification number and a reference to a picture file and properties defining the position of the picture in the template.

Another template 103 comprises an object 104 e.g. in the form of a text element. The object 104 comprises an object identification number and a property defining the position of the text in the template.

5 Moreover, the template comprises an object 105 in the form of a so-called modifying object. The modifying object comprises an identification number, a property defining an identification number of an object to modify, and a property defining modifying a property of the

10 object to modify. For instance the modifying object 105 can comprise a reference to the object 102 and a property modifying the position of the image to the upper right corner of the template 103.

The result of this template arrangement is shown as the

15 document 106.

Moreover, it is shown how other documents 109 and 110 can be created from the same common template 101. However, the different intermediate templates 107 and 108 may modify the appearance of the template 101 in the

20 documents 109 and 110 differently.

Fig. 2 shows a combination of more templates. The system is inter alia expedient in that it allows for use of any desired number of templates. A combination of more templates is shown. The system is inter alia expedient in

25 that it allows for use of any desired number of templates. The combination of template A, C and B is used to produce document E. Object 102 is taken from A modified by C and again modified by B, so the resulting representation of 102 is determined by B. It can also be

30 seen that the normal objects from B and C are also represented in E.

Fig. 3 shows a flow chart for parsing a document based on one template, which is again based on two templates. Section 1 of the flowchart illustrates processing resulting from data in a document 'DOC 1'. Section 2 of the flow chart illustrates processing resulting from data in a template 'TEMP 1' referenced by the document 'DOC 1'. Section 3 of the flowchart illustrates processing resulting from data in the template 'TEMP 2' referenced by the template 'TEMP 1'. Section 4 of the flowchart illustrates processing resulting from data in the template 'TEMP 3' referenced by the template 'TEMP 1'.

The document 'DOC 1' comprises the following code for referencing the template 'TEMP 1':

```
<smil>
      <head>
      ......
      </head>
      <body>
      <include xinclude:href="TEMP 1"/>
      <seq>
      This is the document
      </seq>
      </body>
</smil>
```

In step 301 the document is opened; subsequently, in step 302 a common Parse Tree is created. In step 303 The processor starts reading the document. In step 304 data is read into the Parse Tree. In step 305 a reference to "TEMP 1" is found (e.g. the tag: <include xinclude:href="TEMP 1"/>). In step 306 "TEMP 1" is opened. In step 307 data is read from "TEMP 1" into the Parse Tree. In step 308 a reference to "TEMP 2" is found. In step 309 "TEMP 2" is opened. In step 310 data is read from "TEMP 2" into the Parse Tree. In step 311 "TEMP 2" is closed and the reading continues in "TEMP 1". In step 312 modifying data is read from "TEMP 1" and merged into the Parse Tree. In step 313 data is read from "TEMP 1" into the Parse Tree. In step 314 a reference to "TEMP 3" is found. In step 315

"TEMP 3" is opened. In step 316 data is read from "TEMP
3" into the Parse Tree. In step 317 "TEMP 3" is closed
and reading continues in "TEMP 1". In step 318 modifying
data is read from "TEMP 1" and merged into the Parse

5   Tree. In step 319 "TEMP 1" is closed and reading
continues in "DOC 1". In step 320 modifying data is read
from "DOC 1" and merged into the Parse Tree. In step 321
data is read from "DOC 1" into the Parse Tree. In step
322 the parse tree is rendered to be presented to a user.

10   A modifying object may be exemplified as below:

```
<replace>
        <node id="1234">
                <attr name="LEFT" value="314"/>
                <attr name="TOP" value="17"/>
15          </node>
</replace>
```

where the position of an object with an identification
number of 1234 is defined by the attributes/properties

20   'LEFT' and 'TOP' set to have the values 313 and 17,
respectively. The object can for instance be a picture.

As a special feature a defined set of objects or all
objects may include a tag identifying a property of the
object as a constraint property that specifies a value or

25   a value range modifications are constrained to. This may
be defined as below:

```
<img src="picture.jpg">
        < constraint attr="LEFT"/>

</img>
```

30   where the property 'LEFT' defining the left position of
the picture object 'picture.jpg' is constrained not to be
changed by modifying objects subsequently.

According to the invention any type of electronic object,
comprising multimedia objects, may be included in a

35   document. Such objects may comprise:

Pictures, animations, video clips, sound samples, text, scripts, etc. The objects may include different types of properties comprising position properties, colour properties, shape properties, temporal/timing properties.

5   Fig. 4 shows a pseudo flowchart illustrating the process of parsing a document 401 based on two templates 402 and 403. In this illustration it is shown how mark-up tags of the document and underlying templates are merged during parsing/scanning. The document 401 and the templates 402,

10   403 may comprise ordinary data/presentation/style objects and modifying objects.

Parsing of the document/templates is started in the document 401 at box 1. As a result of the parsing a reference in the document to the template 402 is

15   identified at box 2. This reference routs the parsing to the template 402. In a similar way parsing is routed to template 403. In template 403 all objects are parsed and their properties are registered (box 3). Subsequently, parsing returns to template 402 where properties of

20   objects parsed in template 403 are changed according to properties of modifying objects in template 402. The cloud 404 represents objects with properties parsed in template 403 and the cloud 405 represents modifying objects with properties to modify properties of the

25   objects in the cloud 404.

In template 402 objects and their properties are parsed. Subsequently, these objects and properties can be modified in the document 401 if any modifying objects are found. Objects with properties and modifying properties

30   are indicated with the clouds 406 and 407.

Properties of the document to be rendered are represented by the cloud 408.

- 9 -

Fig. 5 shows a computer system according to the invention. The computer system comprises a user interface 501, a computer network 502, and a computer 503. The computer 503 comprises a central processing unit 504, a memory 506, and a storage unit 505.

The memory 506 is loaded with objects or components for controlling access privileges; for storing changes; for parsing/scanning a document and templates; and for editing documents/templates. The storage 505 comprises documents and templates stored as electronic files.

The user interface can include a display screen for displaying text, pictures, or video; a loudspeaker for playing sounds/music; and a printer for making printed copies of a document/template.

Generally the term 'rendering' is defined to comprise a transformation of the object to a specified medium, comprising display screens, loudspeakers, printers, etc.

The invention can be embodied as a computer program product, as a system, or as a method.

CLAIMS

------

1. A method of generating a document from a plurality of templates, wherein each template has an identifiable object with properties; the method comprising the steps of:

searching a reference to a first template in an electronic document;

parsing the first template to find a reference to a second template;

parsing the second template to find a first object with properties;

reading the properties, and returning to the first template

continuing parsing the first template to find a second object that specifies changes to the first object and change properties of the first object according to properties in the second object specifying the changes;

rendering the first object.

2. A method according to claim 1 wherein the first object is rendered in accordance with the changed properties.

3. A method according to claim 1 or 2 wherein the properties to be changed of the first object are specified by similar properties in the second object.

4. A method according to any of claims 1 to 3 further comprising the step of: constraining properties of objects such that the properties cannot be modified.

5. A method according to any of claims 1 to 4 further comprising the step of: inserting further objects into the document in response to a user interaction.

6. A method according to any of claims 1 to 5 further comprising the step of: copying a document and referenced templates into a set of files.

7. A method according to any of claims 1 to 6 further comprising the step of: merging a document and referenced templates into a single file.

8. A method according to any of claims 1 to 7 wherein the objects comprise multimedia objects.

9. A method according to any of claims 1 to 8 wherein the document and the templates are written in a mark-up language.

10. A computer program product for performing the method as set forth in any of claims 1 to 10 when run on a computer.

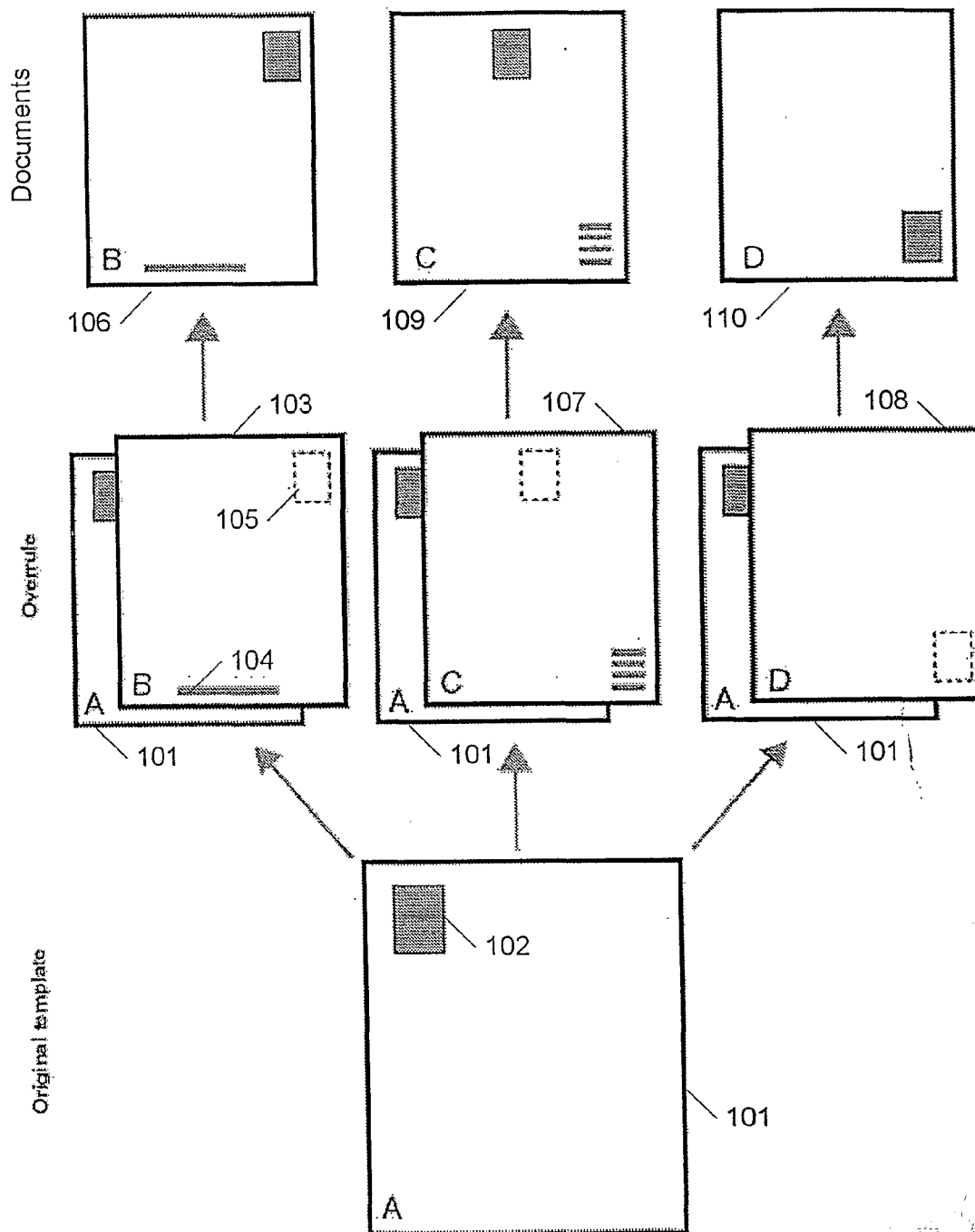11. A computer readable medium for performing the method as set forth in any of claims 1 to 10 when run on a computer.
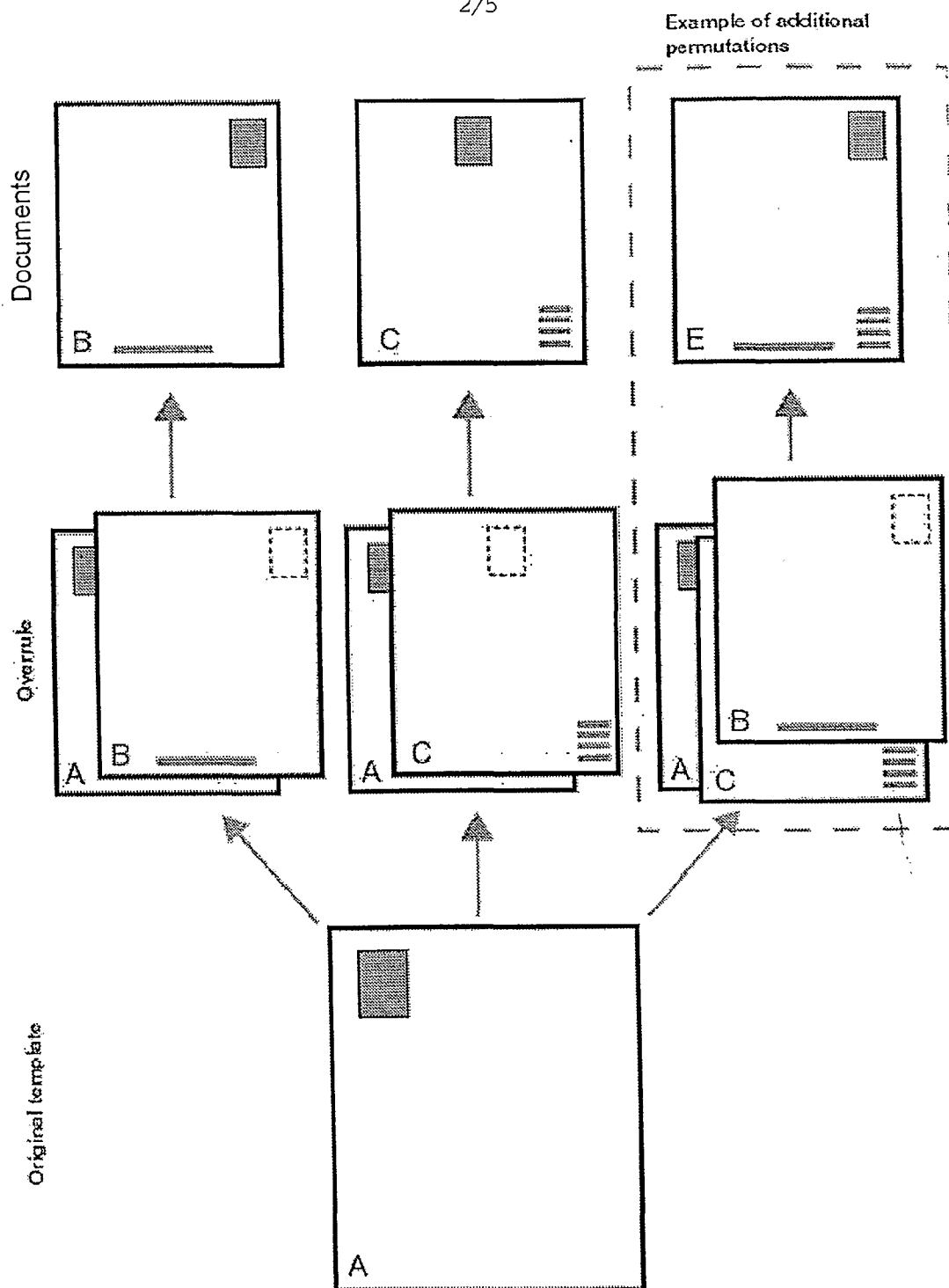
Fig. 1

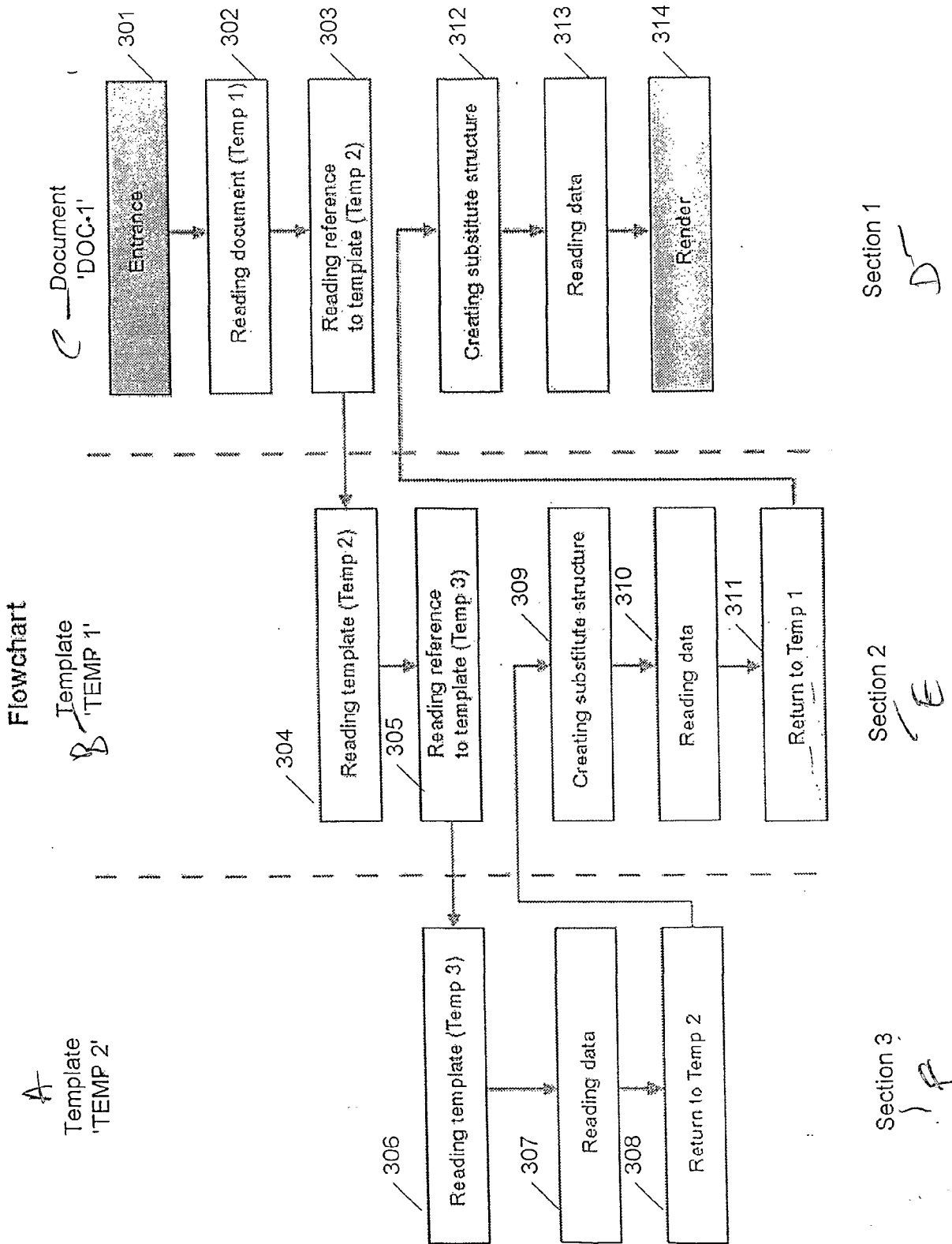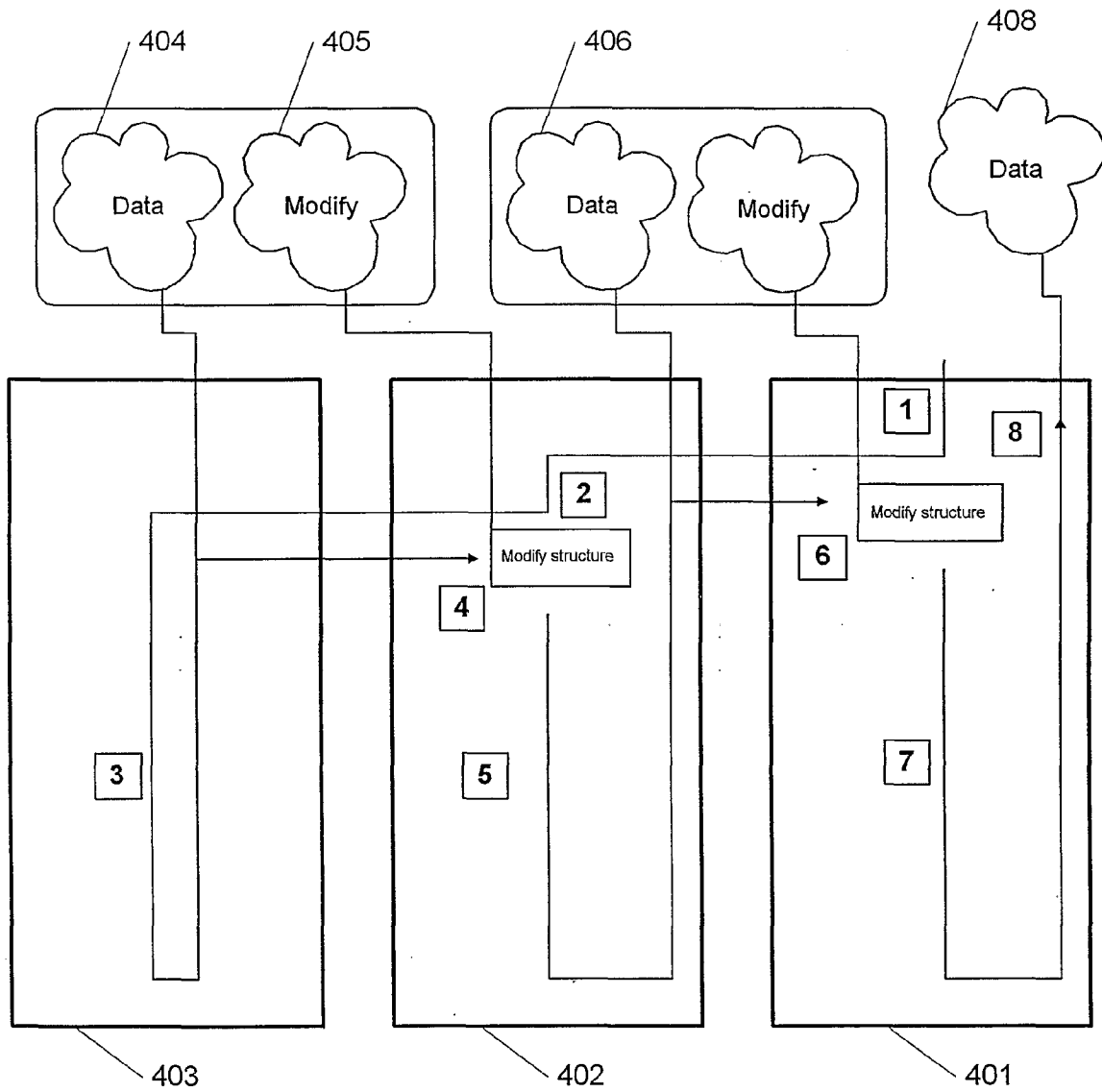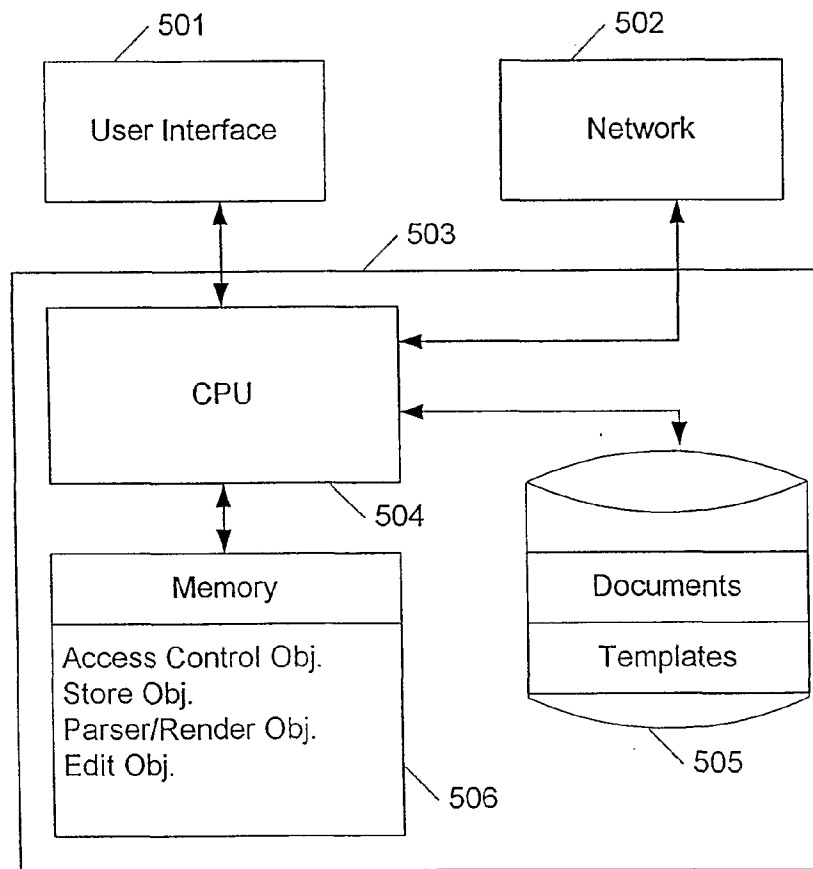Example of additional permutations

Documents

Override

Original template

Fig. 2

FIG. 3

Fig. 4

Fig. 5

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 17/30, G06F 17/21

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6031989 A (CORDELL, JOHN P.), 29 February 2000 (29.02.00), figure 6B, claims 31,38, abstract	1-11
	--	
P,A	WO 0139019 A2 (BITSTREAM INC), 31 May 2001 (31.05.01), page 7; page 20 - page 22; page 65, figures 3,4, abstract	1-11
	--	
A	WO 9108538 A1 (EASTMAN KODAK COMPANY), 13 June 1991 (13.06.91), page 3, line 31 - page 6, line 36, claims 1,4,5,7, abstract	1-11
	--	

[X] Further documents are listed in the continuation of Box C.   [X] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier application or patent but published on or after the international filing date	"X"	document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 December 2001	0 9 -01- 2002

Name and mailing address of the ISA/	Authorized officer
Swedish Patent Office Box 5055, S-102 42 STOCKHOLM	Bo Gustavsson/LR
Facsimile No. +46 8 666 02 86	Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

International application No.

PCT/DK 01/00608

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 9908205 A1 (DIMARCO, CHRYSANNE ET AL.), 18 February 1999 (18.02.99), page 6 - page 13, abstract	1-11

Form PCT/ISA/210 (continuation of second sheet) (July 1998)

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
US	6031989	A	29/02/00	NONE			
WO	0139019	A2	31/05/01	AU	1795501	A	04/06/01
WO	9108538	A1	13/06/91	EP	0462242	A	27/12/91
				US	5181162	A	19/01/93
WO	9908205	A1	18/02/99	AU	8795998	A	01/03/99
				EP	1002284	A	24/05/00
				GB	9716986	D	00/00/00
				GB	9720133	D	00/00/00

9/17/2009, EAST Version: 2.4.1.1